TOMBALA

Design and Requirement Document

Yasemin Alpay

03.01.2016

Introduction

In this project, usual “Tombala/Bingo” game will be implemented using Python. The scenario of the game is given below:

- There will be several users that can connect to the server. Each user can either join a session, or create his/her own new session to start a game.
- Once a user joins or starts a session, he/she will wait for other users to join the game. The game will start in 15 seconds if two users have been acquired at minimum. The maximum number of users that can join the game is five.
- Right after the game starts, server will draw new cards for every user in the session. There will be 3 rows in each card and every row consists of 5 numbers in the range of 1 to 90.
- After each user gets his/her card, server will pick numbers from 1 to 90 and sends this number to every user to check. Once a number is sent, it cannot be sent again.
- User will check the number that has been gotten from server and try to match this number with the numbers in his/her card. If there is a match, he/she will sign the number in the card and will send a signal to the server that he/she is ready for the next number. If there is not any match, he/she will send the signal without signing any number in the card.
- If a user is successful in matching every number in a row, he/she will call for “Çinko” and sends this signal to the server. Server will verify the “Çinko” and sends a signal if “çinko” is true or not. If it is true, this signal is sent to every user in the session with the message “First/Second Çinko by user xxx”.
- If a user makes “Çinko” for every row, he/she can call for “Tombala/Bingo”. This signal is verified again by the server. If it is successful, “Game is over, winner is user xxx” message is sent to every user in the session. Afterwards, session will be ended.

1. Application Protocol

1.1 Messages from Client to Server

New User

New user will be connected to the server with a nickname.

Client request: USR <nickname>

Server response if successful: HEL <nickname>

Server response if user name exists: REJ

Note: Client parser will add username and IP address of the user to a user list.

User Quits Server

Client request: QUI

Server response: BYE

List Sessions

This message will be used for listing all sessions in the server. User will list sessions when he/she enters the game for the first time or when the game is over. This request is also needed to refresh the sessions when a new session is created.

Client request: LSQ

Server response: LSA <colon separated sessions>

Example: LSA session1:session2

List Users

When a user joins a session, he/she may wish to see all users in the session.

Client request: LUQ <session name>

Server response: LUA<colon separated nicknames>

Example: LUA nickname1:nickname2

Join Session

Client request: JOS <session name>

Server response if join successful: JOK

Note: When a join is successful, user will be added to session-user list in the server side. This list holds the information of the users in the particular session.

Create New Session

User creates new session to start a game.

Client request: NES <session name>

Server response if session is created: SOK

Server response if session name already exists: SER

Wait for Users

After the user joins a session or starts a new session, he/she will wait for other users to join. If every user is ready, server will draw cards for each user.

Client request: RDY

Server response if all users ready: NEW <card list with colon separated rows>

Example: NEW 1 45 9 42:15 76 13 74:66 11 37 83

Note: The numbers in the card list will be unique and will be in the range in 1 to 90.

Prompt New Number

User has checked the number in his/her card and waits for a new number.

Client request: NXT

Server response: NUM <random number from 1 to 90>

Example: NUM 56

Çinko

User signed every number in a row in his/her card and calls for “Çinko”.

Client request: CNK <card list, cinko count>

Example: CNK 1 45 9 42:15 76 13 74:66 11 37 83, 1

Server response if çinko true: COK <nickname of user and cinko count>

Example: COK yasemin 1

Server response if çinko not true: CER

Note: User will not be able to call for second Çinko if cinko_count variable is not 1. This variable will be changed only by server and will be passed to client.

Tombala

User signed every number in his/her card and calls for “Çinko”.

Client request: TOM <card list, cinko count>

Example: CNK 1 45 9 42:15 76 13 74:66 11 37 83, 2

Server response if tombala true: TOK <nickname of winner>

Example: TOK yasemin

Server response if tombala not true: TER

Note: User will not be able to call for Tombala if cinko_count variable is not 2. This variable will be changed only by server and will be passed to client.

Game Over

User has viewed the winner message and will be kicked from the session.

Client request: FIN

Server response: END

Wrong Command

User entered a wrong command that cannot be parsed in client parser.

Client request: Wrong command

Server response: ERR

User Not Logged In

User has not logged in yet, and cannot send any commands.

Client request: Command

Server response: ERL

Request	Parameter	Response	Parameter	Definition
USR	nickname	HEL	nickname	User accepted
		REJ		Nickname exists
QUI		BYE		User exits
LSQ		LSA	session1:session2…	List sessions
LUQ	sessionname	LUA	nick1:nick2…	List users in a session
JOS	sessionname	JOK		Joins session
NES	sessionname	SOK		Creates new session
		SER		Session name exists
RDY		NEW	cardlist	Gets new card from server
NXT		NUM	random_number	Gets new number from server
CNK	cardlist,cinko_count	COK	nick:nth cinko	Sends user name who gets "Çinko"
		CER		Invalid "Çinko"
TOM	cardlist,cinko_count	TOK	nickname	Prints winner name
		TER		Invalid "Tombala"
FIN		END		Game over
Command		ERR		Wrong command
Command		ERL		User not logged in

Table 1: Application Protocol Client-Side

1.2 Messages from Server to Client

Connection Test

This message will be sent to clients periodically to check connection status.

Client request: TIC

Server response: TOC

Request	Parameter	Response	Parameter	Definition
TIC		TOC		Connection test
JOK		RDY		User connects the session and ready for new game
SOK		RDY		User creates new session and ready for new game
NEW	cardlist	NXT		User waits for new number
NUM	random_number	NXT		User waits for new number
TOK	nickname: "Tombala"	FIN		Session is closed

Table 2: Application Protocol Server-Side

1.3 User Commands

All interactions between client and server will be within command-line.

User can enter specific commands to play the game:

/nick <nickname> : Changes nickname

/quit : Quits server

/list session : List sessions in the server

/list user : List users in the session which user is playing in right now

/join <session name> : Joins an existing session

/new <session name> : Creates new session

/next : Ready for next number

/close <number> : Sign number in the card

/çinko : Calls for “çinko”

/tombala : Calls for “Tombala”

/help : Prints all commands

1.4 Sample Game

>>/nick yasemin
Hello Yasemin!
>>/list session
Session1
Pros
Newbies
>>/join Pros
Your card is:
1 45 9 42
15 76 13 74
66 11 37 83
New number: 73
>>/next
Your card is:
1 45 9 42
15 76 13 74
66 11 37 83
New number: 76
>>/close 76
Your card is:
1 45 9 42
15 76* 13 74
66 11 37 83
>>/next
Your card is:
1 45 9 42
15 76* 13 74
66 11 37 83
New number: 83
>>/close 83
Your card is:
1 45 9 42
15 76* 13 74
66 11 37 83*
>>/next
…
…
Your card is:
1* 45* 9* 42
15* 76* 13 74*
66 11 37 83*
New number: 88
Ali First Cinko!
>>/next
Your card is:
1* 45* 9* 42
15* 76* 13 74*
66 11 37 83*
New number: 42
>>/close 42
Your card is:
1* 45* 9* 42*
15* 76* 13 74*
66 11 37 83*
>>/cinko
Yasemin First Cinko!
…
…
Your card is:
1* 45* 9* 42*
15* 76* 13* 74*
66* 11 37* 83*
New number: 11
>>/close 11
Your card is:
1* 45* 9* 42*
15* 76* 13* 74*
66* 11* 37* 83*
>>/tombala
Winner is Yasemin!
Game over!
>>